**《数据结构综合设计》实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 人工智能与大数据学院 | 专业 | 虚拟现实技术 | 班级 | 21级3班 | 学生姓名 | 李砚祖 |
| 实验  周次 | 13-14 | 实验  日期 | 2023.5.26 | 学时 | 6 | 教师姓名 | 李昊康 |
| 项目名称 | | 排序的应用 | | | | | |
| 实验  类别 | 🗹验证型实验 🞎设计型实验 🞎综合型实验 🞎其它 | | | | | 成绩：92 | |
| 1. 实验目的及具体要求   实验目的：  1.实现多种类型的排序算法（插入排序、交换排序、选择排序、归并排序等）；  2.理解排序过程；  3.计算比较次数和移动次数，对比分析算法性能的优劣与适用场景；  具体要求：  编写程序实现插入排序、希尔排序、冒泡排序、快速排序、简单选择排序、堆排序。中任意3种。   1. 实验仪器、设备和材料   硬设备：PC机  软件环境：Windows VS2019   1. 实验内容、步骤及实验数据记录   1.插入排序  #include <stdio.h>  void insertionSort(int array[], int n) {  int i, key, j;  for (i = 1; i < n; i++) {  key = array[i];  j = i - 1;  }  array[j + 1] = key;  }  int main() {  int array[] = { 12, 11, 13, 5, 6 };  int n = sizeof(array) / sizeof(array[0]);  insertionSort(array, n);  for (int i = 0; i < n; i++)  printf("%d ", array[i]);  printf("\n");  return 0;  }    2.冒泡排序  #include <stdio.h>  void bubbleSort(int arr[], int n){  int i, j, temp;  for (i = 0; i < n-1; i++){  for (j = 0; j < n-i-1; j++){  if (arr[j] > arr[j+1]){  temp = arr[j];  arr[j] = arr[j+1];  arr[j+1] = temp;  }  }  }  }  int main(){  int arr[] = {64, 34, 25, 12, 22, 11, 90};  int n = sizeof(arr)/sizeof(arr[0]);  bubbleSort(arr, n);  printf("Sorted array: \n");  for (int i=0; i < n; i++){  printf("%d ", arr[i]);  }  return 0;  }    3.希尔排序  #include <stdio.h>  void shellSort(int arr[], int n) {  int gap, i, j, temp;  for(gap = n/2; gap > 0; gap /= 2) {  for(i = gap; i < n; i++) {  temp = arr[i];  for(j = i; j >= gap && arr[j-gap] > temp; j -= gap) {  arr[j] = arr[j-gap];  }  arr[j] = temp;  }  }  }  int main() {  int array[] = { 12, 11, 13, 5, 6 };  int n = sizeof(array) / sizeof(array[0]);  shellSort(array, n);  for (int i = 0; i < n; i++)  printf("%d ", array[i]);  printf("\n");  return 0;  }     1. 选择排序   #include <stdio.h>  void selection\_sort(int arr[], int len) {  int i, j, min\_idx, temp;  for(i = 0; i < len - 1; i++) {  min\_idx = i;  for(j = i + 1; j < len; j++) {  if(arr[j] < arr[min\_idx]) {  min\_idx = j;  }  }  temp = arr[i];  arr[i] = arr[min\_idx];  arr[min\_idx] = temp;  }  }  int main() {  int array[] = { 12, 11, 13, 5, 6 };  int n = sizeof(array) / sizeof(array[0]);  selection\_sort(array, n);  for (int i = 0; i < n; i++)  printf("%d ", array[i]);  printf("\n");  return 0;  }    分析：  冒泡排序：时间复杂度为 O(n^2)。  选择排序：时间复杂度为 O(n^2)。  插入排序：时间复杂度为 O(n^2)。  希尔排序：时间复杂度为 O(nlog^2n)。  以上四种排序算法的空间复杂度均为 O(1)，即都是原地排序算法，不需要额外的空间来存储数据。因此，在空间复杂度方面这四种算法都是比较高效的排序算法。 | | | | | | | |

说明：1. 实验周次：填写实际上课周，如第5-8周上课填“5-8”或第10周上课填“10”。

1. 实验报告各部分内容需详实填写，按实验指导书上的评分标准给出分数。
2. 实验类型参考实验类型说明文件。